南 臺 科 技 大 學

電 子 工 程 系

專題製作報告

題目 :醫療圖像處理

指導教授 :方信普 副教授

班 級 :微電四甲

製作學生 :陳佳萱 學號 :4A03A053

中華民國 104 年 4 月 10 日

|  |
| --- |
| **南臺科技大學電子工程系學生專題報告文責自負聲明書**  本人 陳佳萱 **瞭解並保證所撰專題報告完全遵守著作權法及學術倫理**，師長業已善盡告知、監督暨審查之義務。專題報告**倘有抄襲、改作、妨礙他人著作權，或其他一切有違著作權及學術倫理之情事，及衍生相關民刑事責任者，概由本人負責**，與指導教授、專題考試委員及南臺科技大學電子工程系或其他師長無關。  專題報告題目： 醫療圖像處理  此致  南臺科技大學電子工程系  聲明人： 陳佳萱  學號： 4A03A053  簽名：  中華民國 104 年 4 月 10 日  (請以正楷書寫) |

**目錄**

[第一章、 摘要 7](#_Toc420363170)

[第二章、 簡介 8](#_Toc420363171)

[2.1 研究動機與目的 8](#_Toc420363172)

[數位影像處理的起源 8](#_Toc420363173)

[2-1-1 研究動機 8](#_Toc420363174)

[2-1-2 研究目的 8](#_Toc420363175)

[2.2 預期成果 8](#_Toc420363176)

[第三章、 理論 10](#_Toc420363177)

[3.1 Visual C# 10](#_Toc420363178)

[3.2 AForge 10](#_Toc420363179)

[3.3 Emgu 13](#_Toc420363180)

[3.4 直方圖等化 15](#_Toc420363181)

[3.5 ROI 18](#_Toc420363182)

[第四章、 製作步驟 19](#_Toc420363183)

[4.1 系統架構 19](#_Toc420363184)

[4.2 介面設計 20](#_Toc420363185)

[4.3 程式設計 20](#_Toc420363186)

[4-3-1 選擇圖檔/儲存圖檔 20](#_Toc420363187)

[4-3-2 ROI 23](#_Toc420363190)

[4-3-3 Invert 24](#_Toc420363191)

[4-3-4 Canny 25](#_Toc420363192)

[4-3-5 Sharpen 27](#_Toc420363193)

[4-3-6 Threshold 29](#_Toc420363194)

[第五章、 作品成果 31](#_Toc420363197)

[5.1 成品介面 31](#_Toc420363198)

[第六章、 討論 33](#_Toc420363199)

[6.1 遭遇難題與克服 33](#_Toc420363200)

[第七章、 結論 34](#_Toc420363201)

[第八章、 參考資料 35](#_Toc420363202)

[第九章、 參加競賽成果 36](#_Toc420363203)

[第十章、 專案管理 37](#_Toc420363204)

[第十一章、 電子系學生實務專題(二)評量尺規(Rubrics) 40](#_Toc420363205)

圖目錄

[圖3.1 下載AForge 11](#_Toc420363239)

[圖3.2 解開AForge壓縮檔 12](#_Toc420363240)

[圖3.3 開啟Visual C# 12](#_Toc420363241)

[圖3.4 加入參考 13](#_Toc420363242)

[圖3.5 Emgu CV的基本架構 14](#_Toc420363243)

[圖3.6 路徑參考 15](#_Toc420363253)

[圖3.7 加入環境變數 15](#_Toc420363256)

[圖3.8 圖像灰階後的直方圖 16](#_Toc420363258)

[圖3.9 經等化後的圖像 17](#_Toc420363259)

[圖4.1 系統架構圖 19](#_Toc420363260)

[圖4.2 程式執行時的介面 20](#_Toc420363261)

[圖4.3 呼叫圖檔程式 21](#_Toc420363262)

[圖4.4 點選開即可連結電腦內的資料 21](#_Toc420363264)

[圖4.5 存圖程式碼 22](#_Toc420363266)

[圖4.6 執行中存圖 22](#_Toc420363268)

[圖4.7 ROI區域選擇的程式碼 23](#_Toc420363269)

[圖4.8 Invert程式碼 24](#_Toc420363270)

[圖4.9 Invert結果 25](#_Toc420363271)

[圖4.10 Canny程式碼 26](#_Toc420363272)

[圖4.11 Canny結果 27](#_Toc420363273)

[圖4.12 濾波遮罩 28](#_Toc420363274)

[圖4.13 函數的梯度向量及大小 28](#_Toc420363275)

[圖4.14 Sharpen程式碼 28](#_Toc420363277)

[圖4.15 Sharpen結果 29](#_Toc420363278)

[圖4.16 Threshold結果 30](#_Toc420363279)

[圖5.1 程式執行時介面 31](#_Toc420363280)

[圖5.2 Histogram等化 32](#_Toc420363281)

[圖5.3 整張圖的強化 32](#_Toc420363284)

[圖9.1 參賽證明 36](#_Toc420363285)

1. 摘要

醫學圖像處理一直是在追求如何將圖像加強的更為清楚，使得醫療人員能更輕鬆的獲得資訊來了解病情並制訂醫療方法。

本專題是經由Visual C#、AForge以及Emgu所共同撰寫而成，利用C#為主要程式的架鋼，設計出介面的使用，再由區域選擇(ROI)選取所要加強分析的地方，再利用Visual C#底下的擴充封裝AForge及Emgu為圖像作處理。AForge作常見的圖像加強，分別是Canny、Invert、Sharpen、HistogramEqualization等，Emgu則是用來計算等化圖像的色素，以達到圖像更為明顯。

醫學圖像處理在現今的醫學界的相關應用非常廣泛，開發的技術可以應用於X光片 (Radiography)、超音波 (ultrasound)、斷層掃描 (CT)、核磁共振 (MRI)、正電子斷層掃描(PET)、及單光子斷層掃描 (SPECT) 等影像。

1. 簡介
2. 研究動機與目的

數位影像處理的起源

影像處理的第一個應用之一是在報紙業，當時倫敦和紐約之間透過海底電纜首先傳送圖片。在1920年代初，巴特蘭電纜圖片傳輸系統的誕生，將傳送一張圖片跨越大西洋所需的時間從一個星期減少到3個小時內。為了實現電纜傳輸，先用專門的列印設備將圖片編碼，然後在接收進行圖片重建。就是利用這樣的方法傳送的，並在電報印表機上利用自形模擬半色調圖案將其還原。

在改善這些初期的數位圖片的視覺品質中所遇到的一些初始問題與列印程序的選擇以及強度準位的分佈有關。用於獲得的列印方法在1921年底被放棄而傾向於一種在電報接收終端機上穿孔的那種照相重製技術。

1. 研究動機

醫學影像是以非破壞性的方式觀察體內骨骼與腦內靜態結構的媒體。藉由醫學影像的分析及視覺化，我們可以獲得器官、組織、及神經的特性，提供體內病灶臨床診斷及腦部認知研究之用。

相對地醫學影像在整個醫療診斷過程中佔有越來越重要的地位，當醫學影像分析的越清楚，越夠助醫師從影像中獲得足夠的資訊來了解病情並制訂醫療方法，能縮短很多造成誤判的情況，因此獲得更方便有效之診療成果。

1. 研究目的

醫學影像種類繁多包括了像是X光片 (Radiography)、超音波 (ultrasound)、斷層掃描 (CT)、核磁共振 (MRI)、正電子斷層掃描(PET)、及單光子斷層掃描 (SPECT) 等影像，但這些影像中常會有一些較為細小不易方發的地方，如果我們能將其特定的區域去作影像的強化，使得醫學人員更能簡單的去辨識出來。

1. 預期成果

將醫學影像分析得更佳，使得醫療人員們在辨識上能更容易，並設計簡單的操作介面，讓使用者能更容易上手，除此之外，將程式碼更簡化讓執行檔的檔案大小能更壓縮，使執行程式時間能更短且電腦執行時更為輕鬆。

1. 理論
2. Visual C#

Visual C#是微軟推出的一種基於.NET框架的、物件導向的高階程式語言。Visual C#由C語言和C++衍生而來，繼承了其強大的效能，同時又以.NET框架類別庫作為基礎，擁有類似Visual Basic的快速開發能力。Visual C#由安德斯·海爾斯伯格主持開發，微軟在2000年發行了這種語言。是一種簡潔且類型安全 (Type-Safe) 的物件導向語言，讓開發人員能夠建置各種可以在 .NET Framework 上執行的安全、強固應用程式。 您可以使用 Visual C# 來建立 Windows 用戶端應用程式、XML Web Services、分散式元件、主從式應用程式、資料庫應用程式以及更多程式。 Visual C# 提供進階的程式碼編輯器、便利的使用者介面設計工具、整合式偵錯工具以及許多其他工具，用以簡化根據 Visual C# 語言及 .NET Framework 來開發應用程式的程序。因此選用Visual C#作為主體程式來撰寫。

Visual C#語法具有高度表達能力，同時也是相當簡單並容易學習的語法。 熟悉 C、C++ 或 Java 的任何人員都能立即辨識 Visual C# 的大括號語法。 任何了解上述其中一種語言的開發人員，一般都能在極短的時間內開始使用 Visual C# 進行工作。Visual C# 語法將 C++ 的複雜度簡化了許多，同時提供強大的功能，例如可為 Null 的實值類型 (Value Type)、列舉類型 (Enumeration)、委派 (Delegate)、Lambda 運算式及直接記憶體存取，而這些都是 Java 沒有的功能。 Visual C# 支援泛型方法和類型 (會提供增強的類型安全 (Type Safety) 和效能) 以及迭代器 (可讓集合類別 (Collection Class) 的實作器定義自訂反覆運算行為，由用戶端程式碼輕鬆運用)。 Language-Integrated Query (LINQ) 運算式會將強類型查詢當成第一級語言建構。

Visual C# 是物件導向語言，因此支援封裝 (Encapsulation)、繼承 (Inheritance) 和多型 (Polymorphism) 的概念。 所有的變數和方法，包括 Main 方法，也就是應用程式的進入點 (Entry Point)，都封裝在類別定義之內。 類別可能直接從一個父類別繼承，不過可以實作任何數目的介面。

1. AForge

AForge.NET 是一個專門為開發者和研究者基於Visual C#框架設計的，包括計算機視覺與人工智能，圖像處理，神經網絡，遺傳算法，機器學習，機器人等領域。這個框架由一系列的類庫和例子組成。其中包括的特征有：

　　AForge.Imaging -一些日常的圖像處理和過濾器。

　　AForge.Vision -計算機視覺應用類庫。

　　AForge.Neuro -神經網絡計算庫。

　　AForge.Genetic -進化算法編程庫。

　　AForge.MachineLearning -機器學習類庫。

　　AForge.Robotics -提供一些機器學習的工具類庫。

　　AForge.Video -一系列的視頻處理類庫。

在本專題中將會使用到AForge.Imaging、AForge.Math這兩種作為影像處理的參考，AForge.Imaging作用於基本的灰階化，二值化還有其他較為簡單的強化功能，而AForge.Math則是用於撰寫ROI的參考。

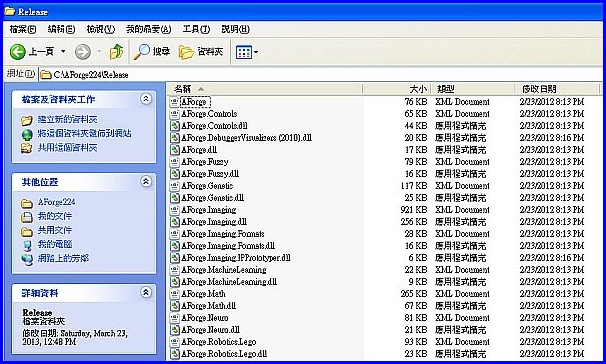
AForge安裝方法：

步驟 1：在網站下載 AForge.NET Framework-2.2.4.zip，下載後檔案大小 34.8MB，解壓後檔案大約 61.7MB。



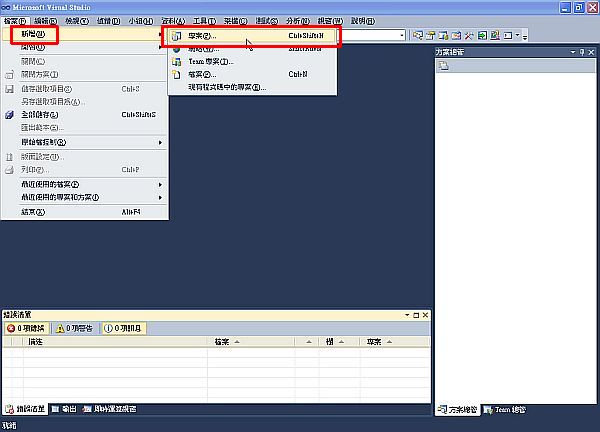
1. 下載AForge

步驟 2：將下載 AForge.NET Framework-2.2.4.zip 的檔案解壓到 C:\ AForge224。



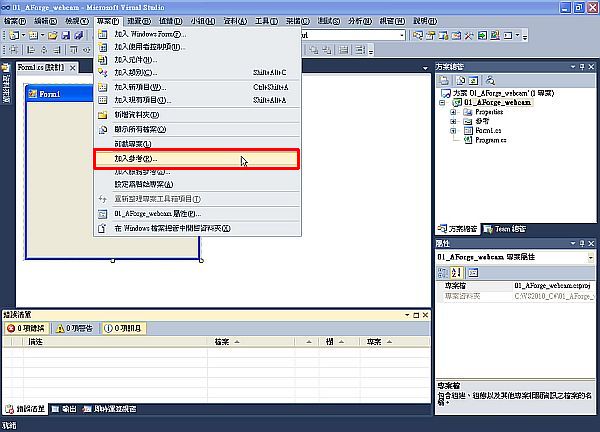
1. 解開AForge壓縮檔

步驟 3：在 Visual Studio 2010 中建立 Visual C# Windows Form 應用程式。



1. 開啟Visual C#

步驟4：專案屬性的參考路徑加入。

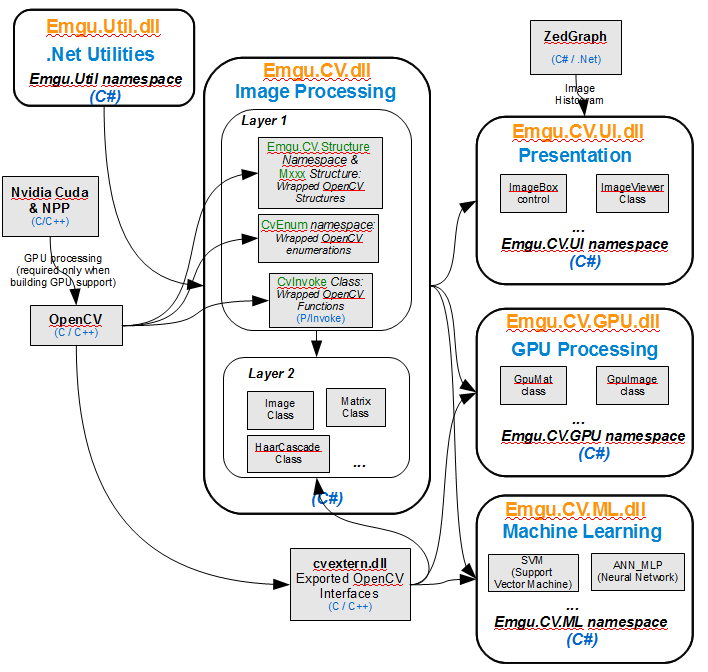


1. 加入參考
2. Emgu

Emgu CV是.NET平台下對OpenCV圖像處理的封装，也就是.NET版。可以運行在Visual C#、VB、VC++等。Emgu CV 有兩層封裝，如下：

第一層是基本層，直接對應OpenCV.包括函數、結構和枚舉類型

第二層是擴充層，包括與.NET庫的一些混合類，適合.NET的優勢。下圖是對Emgu CV的基本架構，我們在專題的配置過程中，也用到了下面的庫文件，在這裏我們可以看到他們的具體作用和架構。此擴充的封裝將運用於醫療圖像的直方圖的計算、顯示以及等化。

1. Emgu CV的基本架構

安裝Emgu CV方法如下：

1. 下載EmguCV安裝檔。
2. 加入參考路徑。

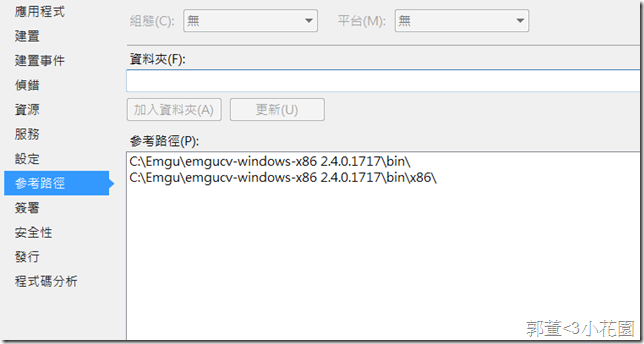
這邊開始會比較不一樣，有人是此須加在專案屬性中的參考路徑加入，而有人則需要在環境變數加入路徑才能執行。

把下面這兩個路徑加入到

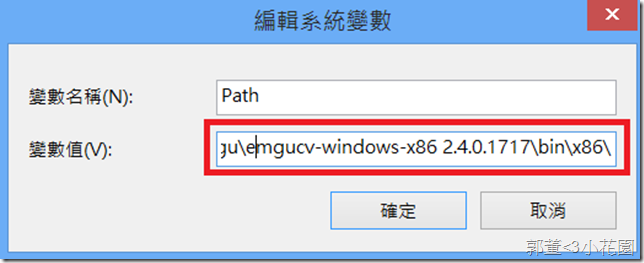
C:\Emgu\emgucv-windows-x86 2.4.0.1717\bin\x86\

C:\Emgu\emgucv-windows-x86 2.4.0.1717\bin\

1. )專案屬性的參考路徑加入。

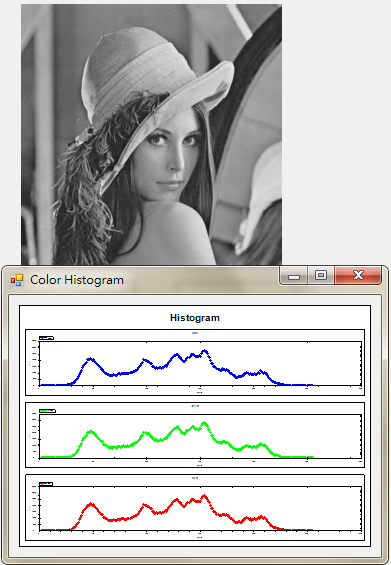


1. 路徑參考
2. )如果不行的話再換成，加入到環境變數Path。



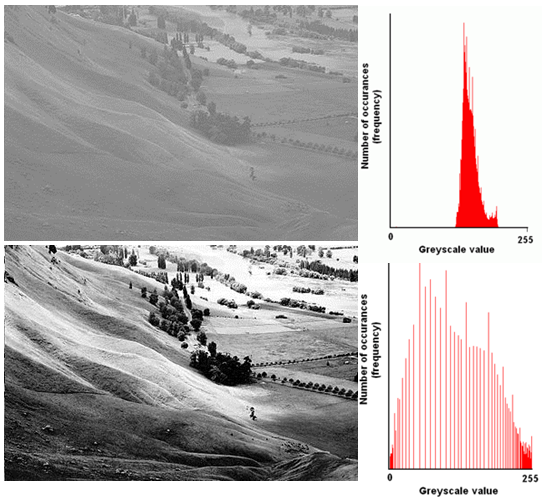
1. 加入環境變數
2. 把要使用到了dll加入至參考。
3. 直方圖等化

直方圖(Histogram)是一種統計圖資料，在影像處理中最常被用來統計一張圖像或是感興趣(ROI)區域的色彩分布。圖像直方圖，是用以表示[數字圖像](http://zh.wikipedia.org/wiki/%E6%95%B0%E5%AD%97%E5%9B%BE%E5%83%8F)中亮度分布的直方圖，標繪了圖像中每個亮度值的[像素](http://zh.wikipedia.org/wiki/%E5%83%8F%E7%B4%A0)數。可以藉助觀察該直方圖了解需要如何調整亮度分布。這種直方圖中，[橫坐標](http://zh.wikipedia.org/wiki/%E7%AC%9B%E5%8D%A1%E5%B0%94%E5%9D%90%E6%A0%87%E7%B3%BB)的左側為純黑、較暗的區域，而右側為較亮、純白的區域。因此，一張較暗圖片的圖像直方圖中的數據多集中於左側和中間部分；而整體明亮、只有少量陰影的圖像則相反。以下可以看到一張圖灰階化之後的直方圖，在圖像作灰階後的處理之後，組成圖像的三原色R(Red)、G(Green)、B(Blue)皆會是相同的。

可以看到一張圖灰階化之後的直方圖，在圖像作灰階後的處理之後，組成圖像的三原色R(Red)、G(Green)、B(Blue)皆會是相同的。

1. 圖像灰階後的直方圖

顏色直方圖本身可以針對任意色彩空間使用，當某圖的色素過於集中在某一區塊時(圖3.3)，在做過處理之後圖像變得較為清晰，且可以看到色素的分布變的平均。

****

1. 經等化後的圖像

因此我們將此功能用於在此專題當中，此專題中我們將直方圖作為八等分去等化，而如何去計算色素所出現的機率可由以下公式得知。讓n_i表示灰階i出現的次數，這樣圖像中灰階為i 的像素的出現機率是p_x(i) = \frac{n_i}{n}, i\in {0,..., L - 1} 

L 是圖像中所有的灰階數，n 是圖像中所有的像素數, p 實際上是圖像的直方圖，歸一化到 0..1。

把 c 作為對應於 p 的累計機率函數, 定義為：

 c(i) = \sum_{j=0}^i p_x(j)

c 是圖像的累計歸一化直方圖。

我們創建一個形式為  y = T(x) 的變化，對於原始圖像中的每個值它就產生一個 y，這樣 y 的累計機率函數就可以在所有值範圍內進行線性化，轉換公式定義為：

y_i = T(x_i) = c(i)

注意 T 將不同的等級映射到 {0..1} 域，為了將這些值映射回它們最初的域，需要在結果上應用下面的簡單變換：

y_i' = y_i \cdot(max - min) + min

上面描述了灰階影像上使用直方圖均衡化的方法，但是通過將這種方法分別用於圖像RGB顏色值的紅色、綠色和藍色分量，從而也可以對彩色圖像進行處理。

1. ROI

感興趣區（通常縮寫ROI，Region of interest），為樣本確定用於特定用途的數據集內的選擇的子集。一個ROI的概念通常在許多應用領域中使用。例如，在[醫學成像](http://en.wikipedia.org/wiki/Medical_imaging" \o "醫療成像)中，腫瘤的邊界，可以在圖像中或在卷中定義，用於測量其尺寸的目的。心內膜邊界可以在圖像上定義的，也許是在心動週期的不同階段，例如最終收縮和舒張末期，以評估心臟功能的目的。在[地理信息系統](http://en.wikipedia.org/wiki/Geographical_information_system)（GIS），一個投資回報率，可以從字面上理解為從2D地圖多邊形選擇。在[計算機視覺](http://en.wikipedia.org/wiki/Computer_vision" \o "計算機視覺)和[光學字符識別](http://en.wikipedia.org/wiki/Optical_character_recognition)的ROI定義了[邊界](http://en.wikipedia.org/wiki/Edge_detection" \o "邊緣檢測)所考慮的對象。

感興趣的區域的例子：

* 1D數據集：上的波形的時間或頻率間隔
* 2D數據集：一個對象的圖像上的邊界
* 3D數據集：在卷的輪廓或曲面勾勒物體（有時也被稱為利益（VOI音量））
* 4D數據集：一個對象的處或在一個時間量的特定時間間隔期間的輪廓

ROI是註解的一種形式，通常具有分類或定量信息相關聯（例如，如體積或平均強度測量），表示為文本或以結構化形式。

有三種完全不同的編碼ROI的方式：

* 作為採樣數據集的一個組成部分，具有獨特的或掩蔽值，可能會或可能不會被外界通常出現的值的正常範圍和哪些標籤個體數據信元。
* 作為單獨的，純粹的圖形信息，例如與載體或位圖（光柵）的繪圖元素，或許與一些伴隨的普通[（非結構）](http://en.wikipedia.org/wiki/Unstructured_data" \o "非結構化數據)中的數據本身的格式文本。
* 作為一個獨立的結構化[的語義](http://en.wikipedia.org/wiki/Semantic" \o "語義)信息（例如編碼[值類型](http://en.wikipedia.org/wiki/Value_type" \o "值類型)）與一組[空間](http://en.wikipedia.org/wiki/Spatial_coordinates" \o "空間坐標)和/或時間的坐標。

1. 製作步驟
2. 系統架構

選取圖片

選取ROI

灰階化

ROI區域的加強(色素等值化、邊緣化、Invert 、HistogramEqualization…等)

顯示

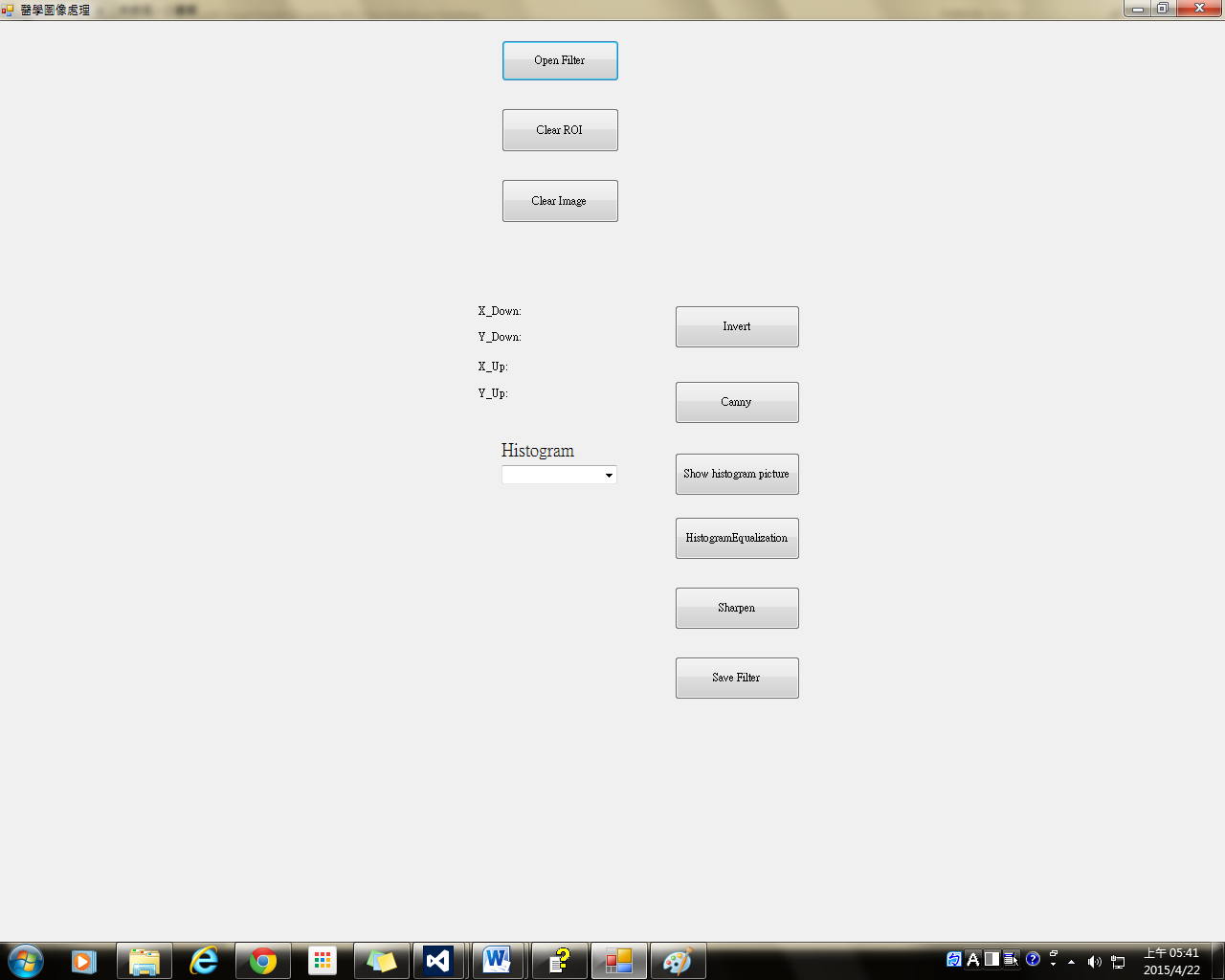
儲存

清除ROI

清除圖片

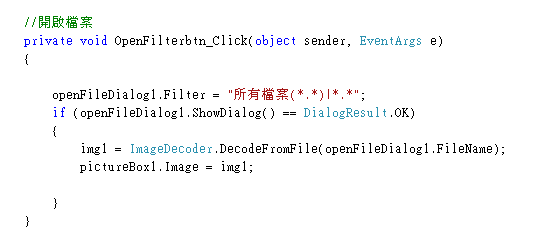
1. 系統架構圖
2. 介面設計

在Visual C#專案中有許多不同的類型可以做選擇，而此專題中使用了Windows Form應用程式，表單(Form)是視窗應用程式中最重要的一個容器之一，也就是我們所謂的視窗，我們設計了三個Picture Box來顯示圖片，分別是選擇圖像、ROI區域圖像以及處理過後的圖像，以及數個Button來做為點選處理圖像加強的按鍵。

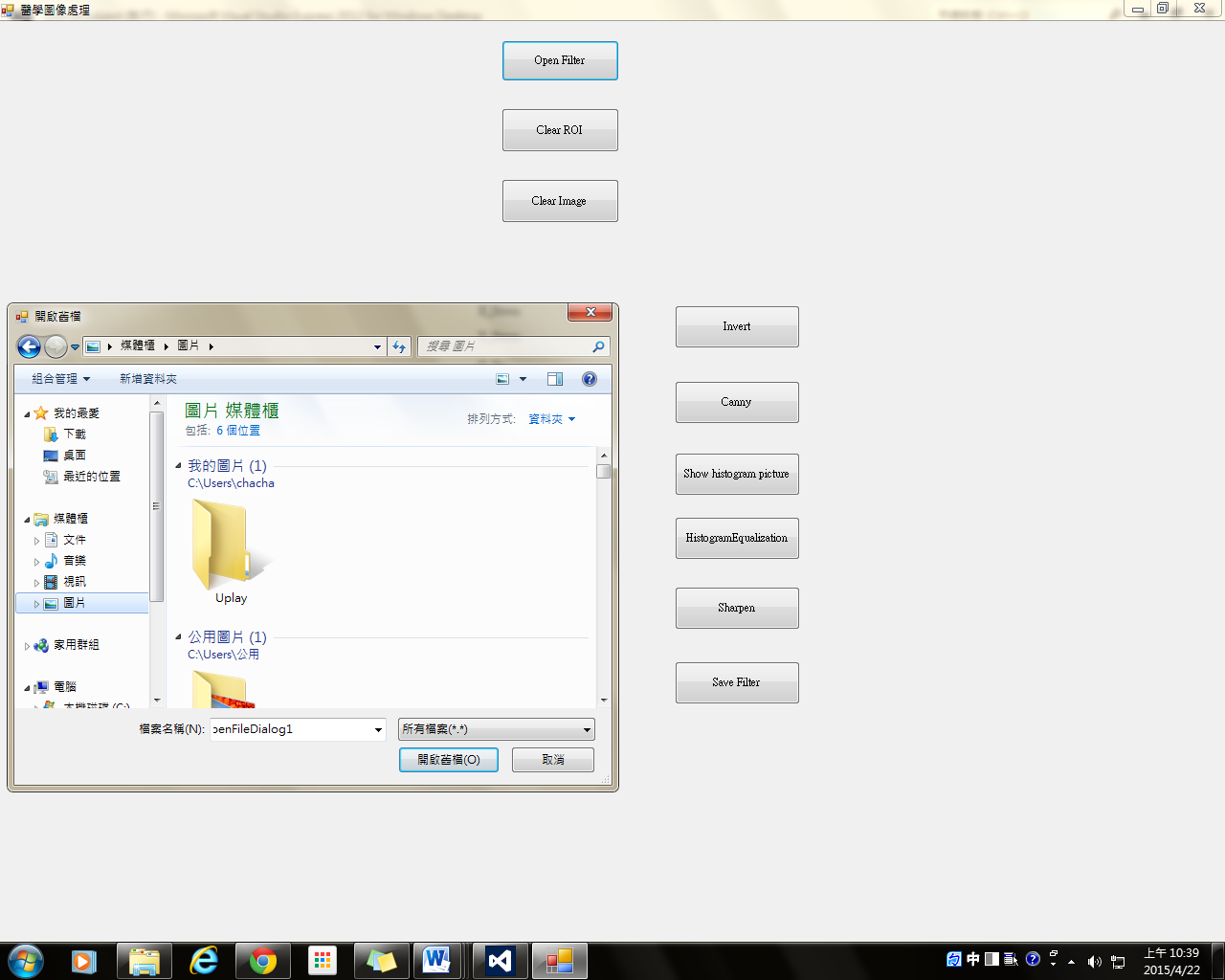


1. 程式執行時的介面
2. 程式設計
3. 選擇圖檔/儲存圖檔

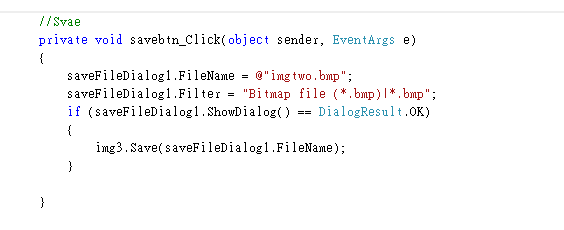
先用Visual C#將介面設計好之後，再來是選擇圖檔，在Visual C#底下電腦連結做得很完善，只要在設計界面先加入openFileDialog在程式內呼叫就好。而儲存檔案也都是相同的原理，在設計頁加入saveFileDialog後，回到程式碼撰寫所要儲存的圖像變數並呼叫，即可以儲存檔案。



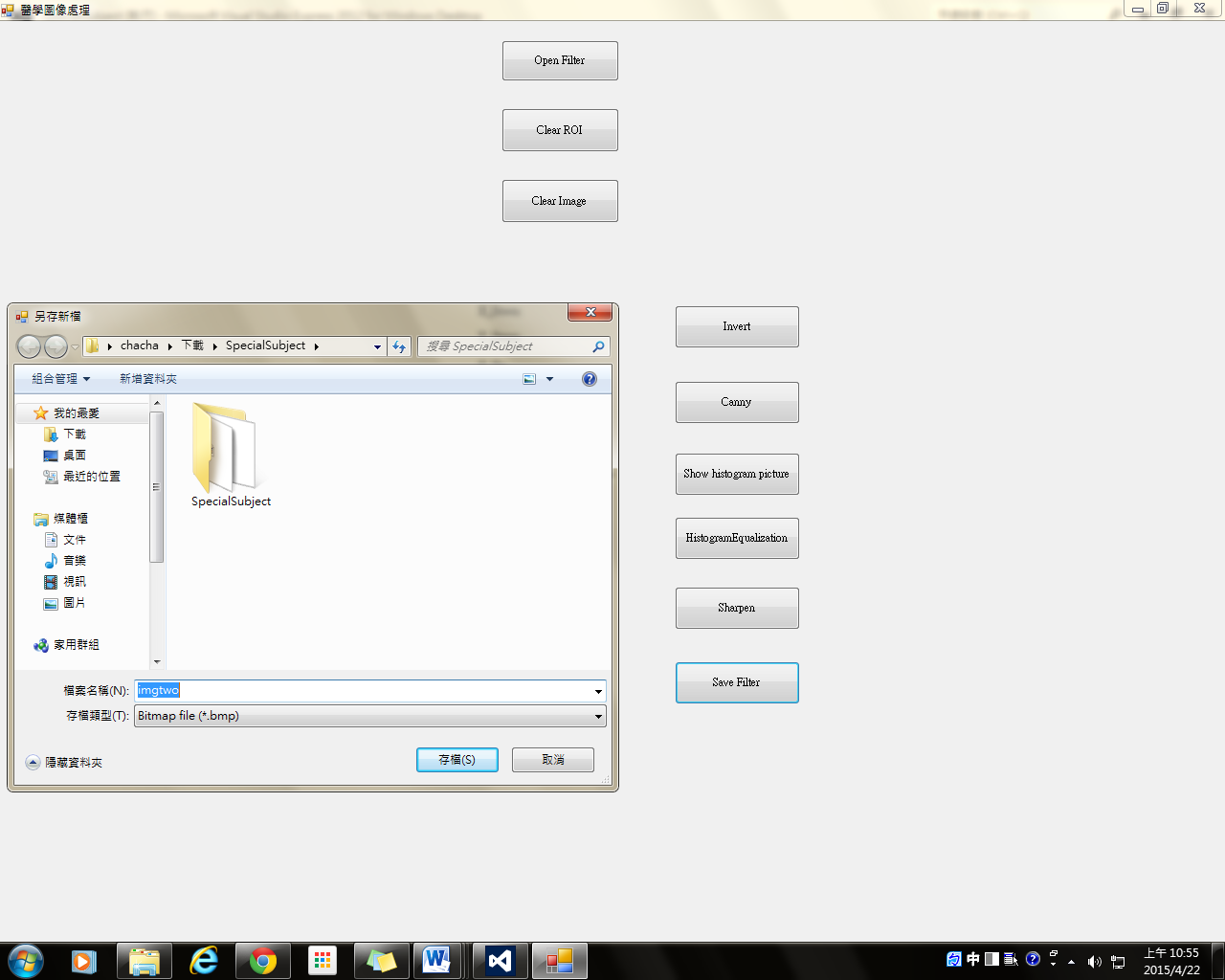
1. 呼叫圖檔程式



1. 點選開即可連結電腦內的資料

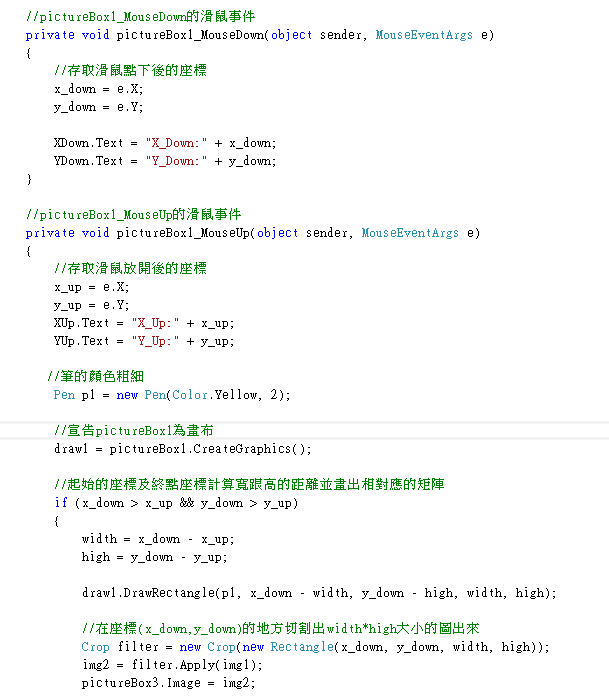


1. 存圖程式碼



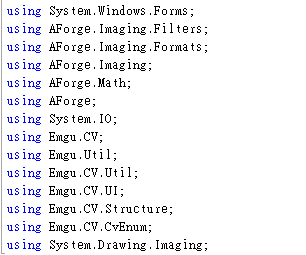
1. 執行中存圖
2. ROI

利用C#內的繪圖物件來設計區域選擇(ROI)，一開始先選擇所需要的圖像，然後再顯示圖像上利用滑鼠事件來觸發接下來的程式，畫好ROI後將可以繼續做之後的圖像強化(圖4.3)。

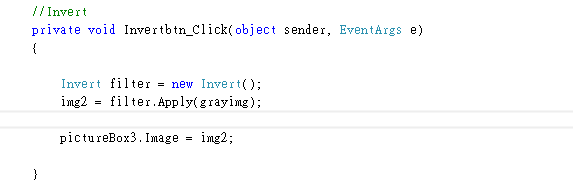


1. ROI區域選擇的程式碼
2. Invert

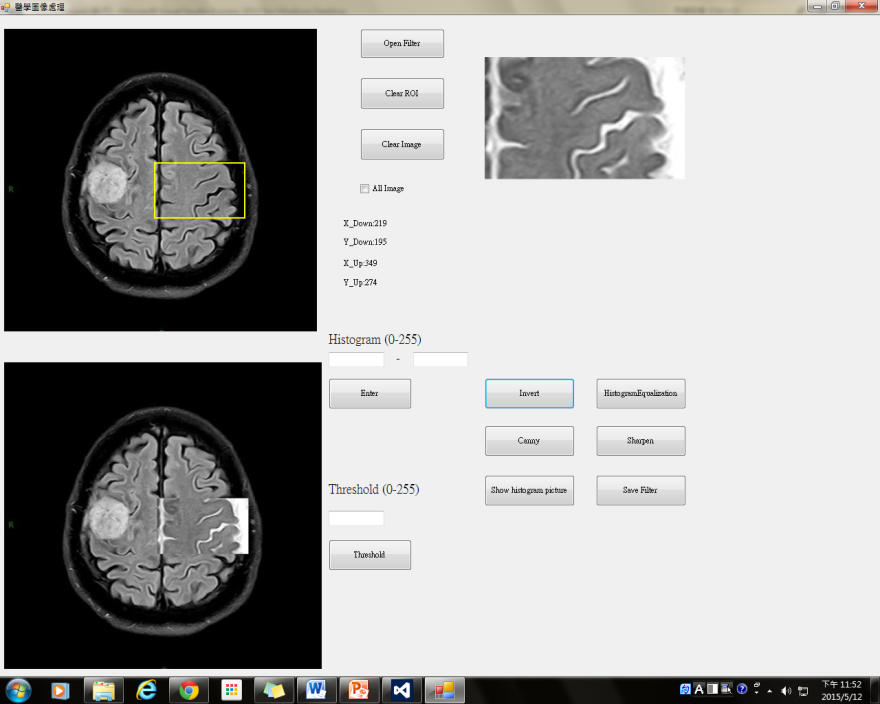
圖像的區域選擇結束後將可以點選Button來作ROI圖像加強，這時加入AForge參考來設計接下來所需要的加強效果，先設計負片效果(Invert)。原理很簡單也就是將 255- 原本的值這樣 0(黑) 就會變成 255(白) 反之 255(白) 就會變成 0(黑)就可以拿到反轉值。



加入參考後的宣告



1. Invert程式碼



1. Invert結果
2. Canny

Canny[邊緣檢測](http://zh.wikipedia.org/wiki/%E8%BE%B9%E7%BC%98%E6%A3%80%E6%B5%8B)運算元是[John F. Canny](http://zh.wikipedia.org/w/index.php?title=John_F._Canny&action=edit&redlink=1)於1986年開發出來的一個多級邊緣檢測算法。較高的亮度梯度比較有可能是邊緣，但是沒有一個確切的值來限定多大的亮度梯度是邊緣多大又不是，所以Canny使用了[滯後](http://zh.wikipedia.org/wiki/%E6%BB%9E%E5%90%8E" \o "滯後)閾值。

滯後閾值需要兩個閾值——高閾值與低閾值。假設圖像中的重要邊緣都是連續的曲線，這樣我們就可以跟蹤給定曲線中模糊的部分，並且避免將沒有組成曲線的噪聲像 素當成邊緣。所以我們從一個較大的閾值開始，這將標識出我們比較確信的真實邊緣，從這些真正的邊緣開始在圖像中跟蹤整個的邊緣。在跟蹤的時候，我們使用一個較小的閾值，這樣就可以跟蹤曲線的模糊部分直到我們回到起點。

一旦這個過程完成，我們就得到了一個二值圖像，每點表示是否是一個邊緣點。一個獲得亞像素精度邊緣的改進實現是在梯度方向檢測二階方嚮導數的過零點

L_x^2 \, L_{xx} + 2 \, L_x \,  L_y \, L_{xy} + L_y^2 \, L_{yy} = 0,

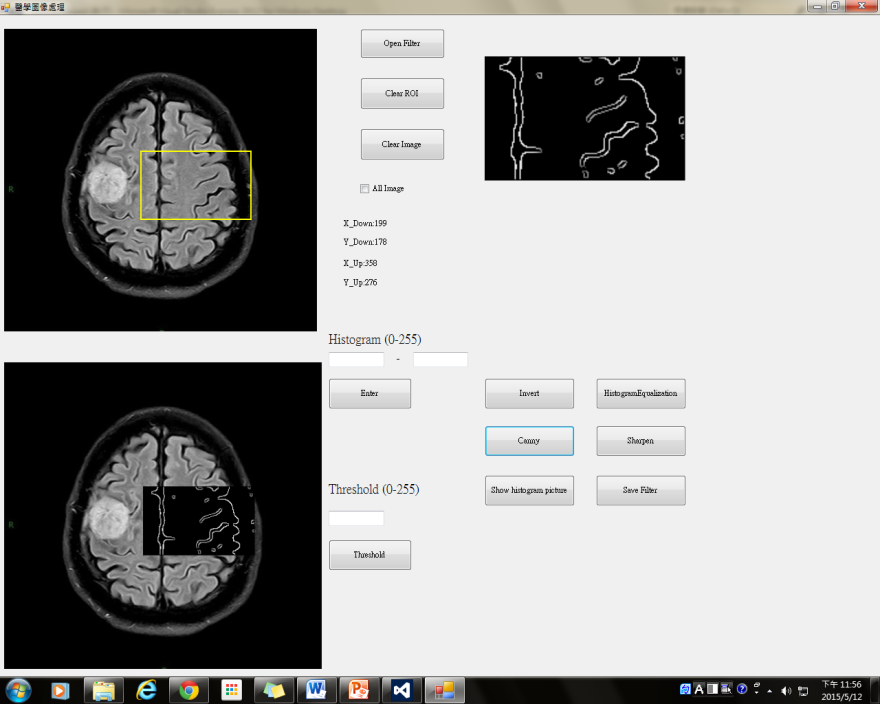
它在梯度方向的三階方嚮導數滿足符號條件

L_x^3 \, L_{xxx} + 3 \, L_x^2 \, L_y \, L_{xxy} + 3 \, L_x \, L_y^2 \, L_{xyy} + L_y^3 \, L_{yyy} < 0

其中L_x, L_y ... L_{yyy}表示用高斯核平滑原始圖像得到的[尺度空間](http://zh.wikipedia.org/w/index.php?title=%E5%B0%BA%E5%BA%A6%E7%A9%BA%E9%97%B4&action=edit&redlink=1" \o "尺度空間 (頁面不存在))表示L計算得到的偏導數。用這種方法得到的邊緣片斷是連續曲線，這樣就不需要另外的邊緣跟蹤改進。滯後閾值也可以用於亞像素邊緣檢測。為了加強Canny後的效果更好，因此我先將圖像二值化後再銳利化，使得邊緣化的效果能更好。



1. Canny程式碼



1. Canny結果
2. Sharpen

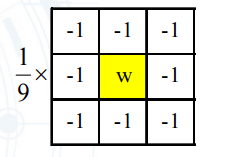
銳化濾波器(Sharpen)主要是用來強化影像的高頻部份，或者是用來強化物體的邊界線，因此會增強影像中的微細部分或恢復己經模糊的細節(圖4.8)。常見的銳化濾波器 ：

* 高通濾波器

高通濾波器是一個用來銳化影像的濾波器，也就是將影像灰階變化大的部分提高其響應，灰階變化平滑的部分降低其響應， 此種濾波器會將影像邊緣位置強化出來。

* 高增幅濾波器

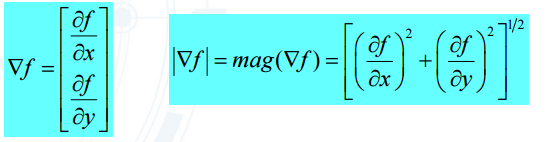
高增幅濾波器所使用的濾波遮罩為(如圖4.8)，其中正中間遮罩係數w之計算方式為 w =9A-1，A≥1；其餘遮罩之係數皆為-1。



1. 濾波遮罩

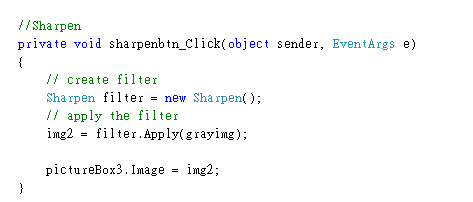
* 導數濾波器

微分對影像的效果是使影像銳化，最常用的 微分方法則是梯度運算，函數的梯度向量及大小可以表示如下：

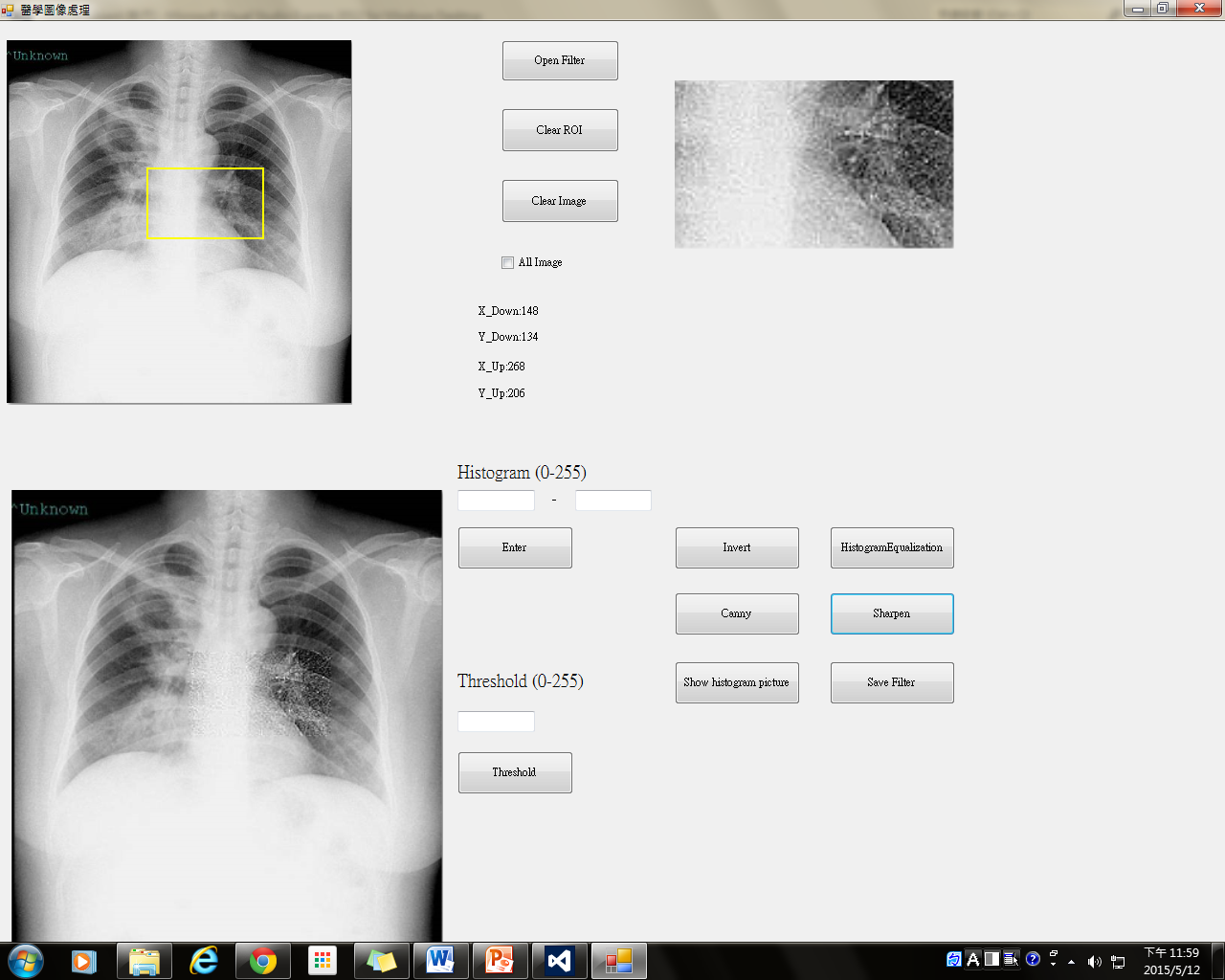


1. 函數的梯度向量及大小

由於上式在使用上並不容易，因此將梯度向量公式轉換成遮罩使用，幾種常見的梯度遮罩包括Prewitt及Sobel，不同的梯度遮罩所產 生的銳化效果也就有所不同。使用上是利用遮罩運算所得到的結果進行判 別，當梯度值大於或等於某定值時，就設定 此點灰階值為255；反之，則以原來的灰階值代之，但如果使用者只在乎影像上的邊線位置時，則可以灰階值0代之。



1. Sharpen程式碼



1. Sharpen結果
2. Threshold

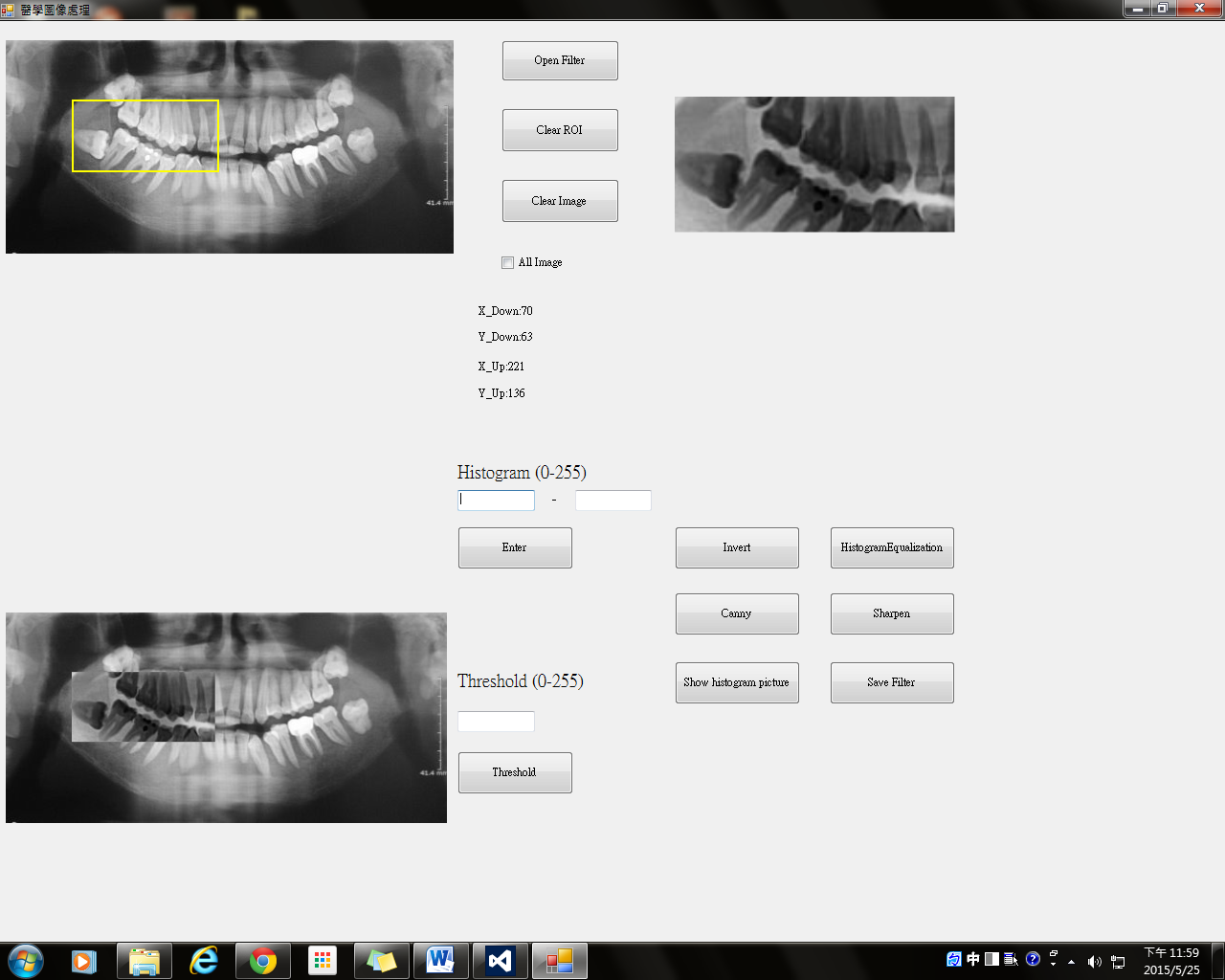
Threshold是一種非線性技術，它計算簡單、去噪效果良好，獲得了廣泛的應用。該方法的關鍵步驟是對分解後的小波係數進行threshold處理，得到小波係數估計。該方法的關鍵點在於如何確定threshold和選取什麼樣的threshold函式，二者的選擇會影響著圖像的去噪效果。

含有雜訊的原始圖像經過小波轉換後，圖像的能量主要集中在小波domain的低頻區域，該區域反映了圖像的基本輪廓，圖像的細節主要分佈在高頻區域，而含有白色雜訊的圖像經過小波轉換後，大部分能量分佈在各個高頻子帶上，小波去噪可以在小波係數的高頻區域內進行。在高頻區域中，圖像的細節資訊主要分佈在幅值較大的小波係數中，雜訊主要集中在幅值較小的小波係數中，而且那些絕對值較大的小波係數含雜訊成分較少，那些絕對值較小的小波係數含雜訊成分較多。所以，當我們對小波domain中高頻子帶上的係數進行取捨時，應該保留那些含有圖像細節資訊的小波係數，去除那些含有雜訊的小波係數。在高頻子帶上，雜訊在統計上是均勻分佈於小波係數中的，因而可以得知那些少數絕對值較大的重要的小波係數一定含有圖像的奇異特性，所以處理時應該舍取或減少絕對值較小的不重要小波係數，保留重要係數。這樣既消減了圖像的雜訊又不會引起圖像的邊緣模糊。因此可以設置一個threshold限制，對於那些大於該threshold的小波係數，認為它們的主要成分為有用訊號，給予收縮後保留；對於那些小於該threshold的小波係數，認為它們的主要成分為雜訊，給予去除，這樣就達到了圖像去噪的目的。



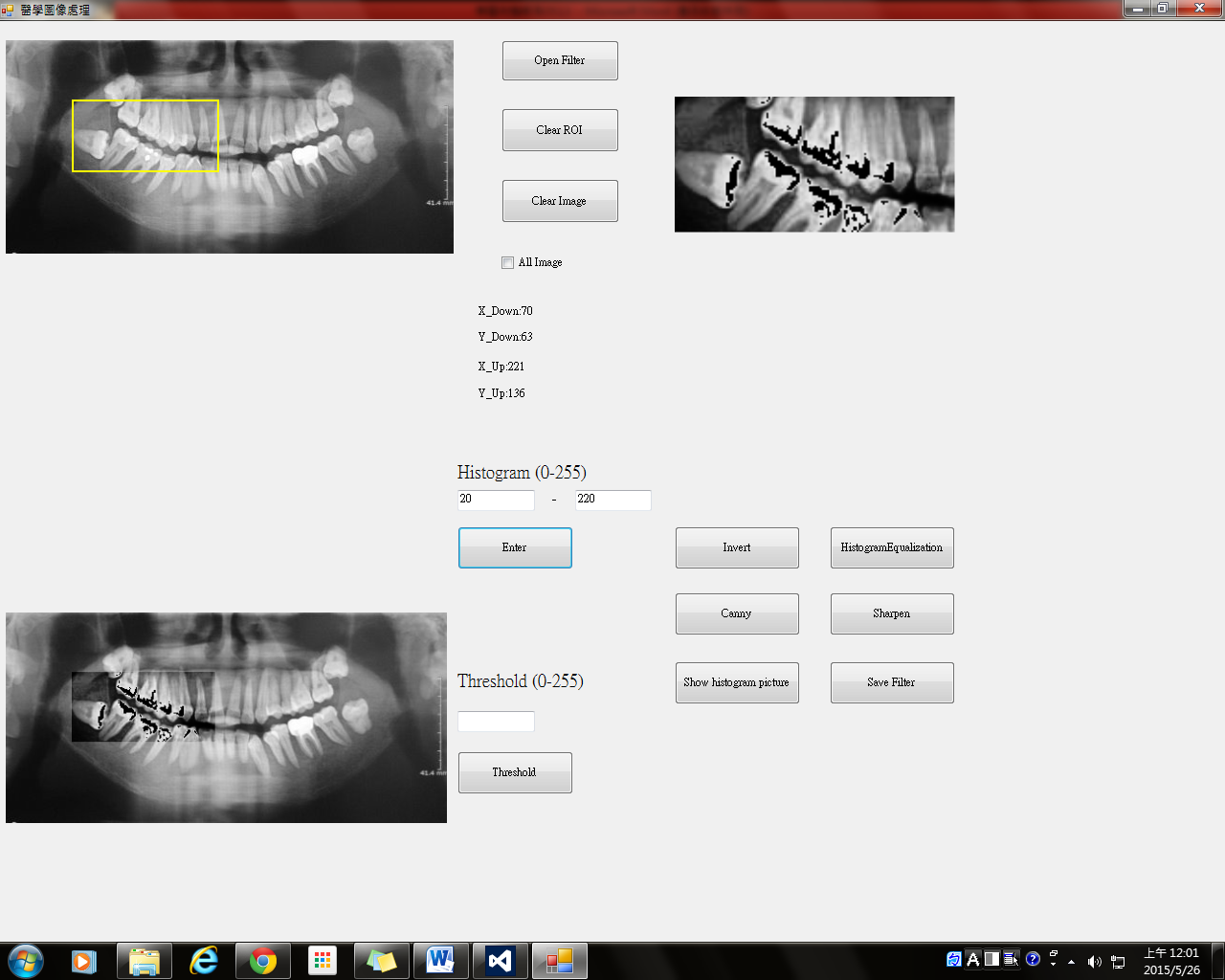
1. Threshold結果
2. 作品成果
3. 成品介面

圖5.1為本專題設計好之後的介面圖，此專題是以Visual C#為主體設計出來再以AForge以及Emgu兩種作為擴充的封裝，作為圖像處理的基準。先選取一張圖象，畫上ROI後將會看到所框選到的影像，接下來將可以對影像作加強處理。如要處理新一處的ROI則可以先點選Clear ROI，清除後再於圖像上畫新的ROI即可。

****

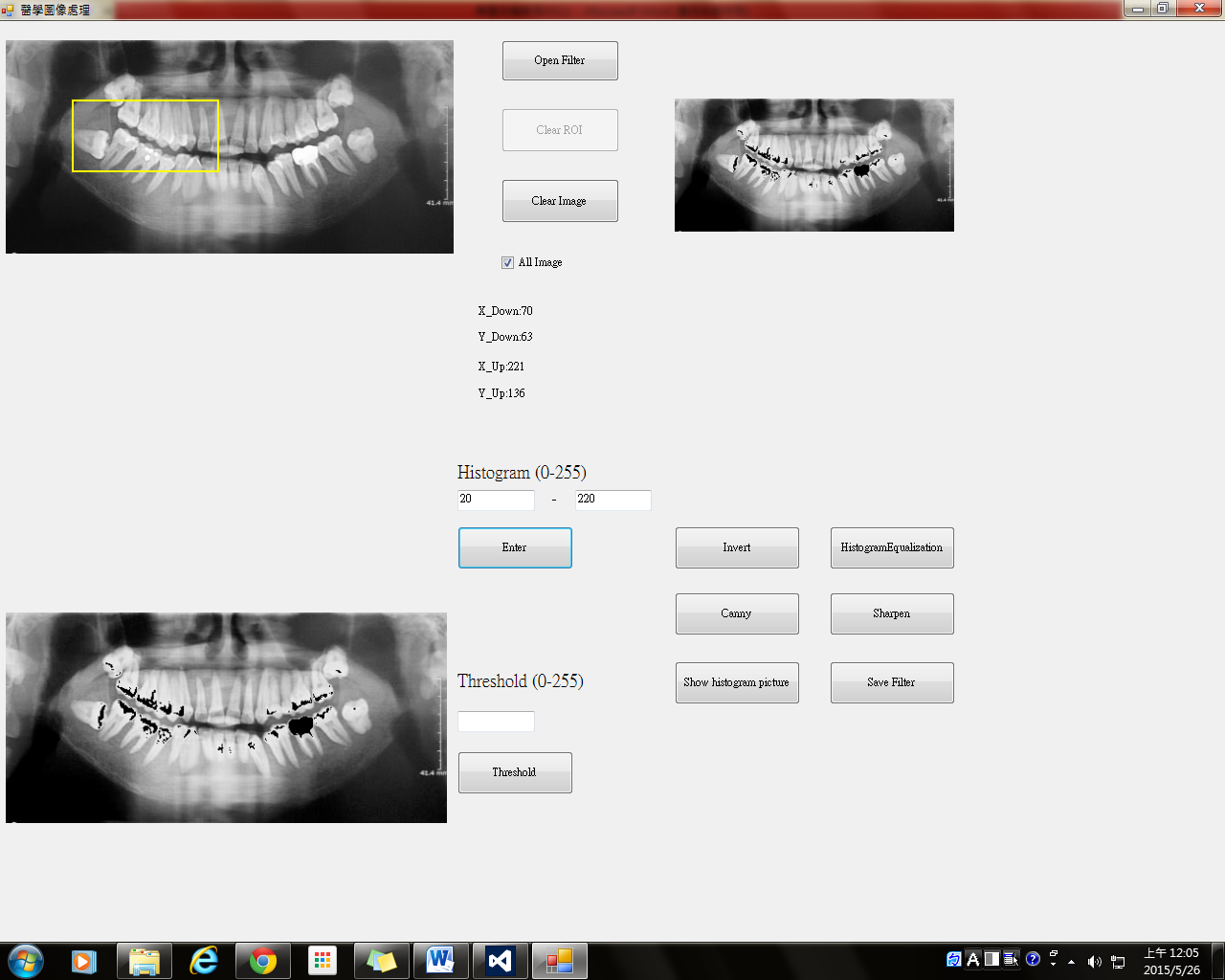
1. 程式執行時介面

在執行程式可看到(圖5.2)中以紅色框標記，可以在輸入列中輸入0~255的像素等化，依照你所需要的需求去作調整。

****

1. Histogram等化

如果想對整張圖作強化的話，就不用在圖像上畫取ROI，可以直接勾選All Image就能將整張圖作強化。



1. 整張圖的強化
2. 討論
3. 遭遇難題與克服

* **問題一：語法的混亂**

**造成原因：**在撰寫程式時，雖然Emgu和AForge都是Visual C#底下擴充的封裝，但由於語法以及變數規格都不同，造成在撰寫上轉換有點困難。

**解決方法：**增設新的變數值，但為了避免變數太多而造成混亂，僅將要作加強的地方增設變數。

* **問題二：無法八等分等直化**

**造成原因：**Emgu和AForge底下的等直化都已經是封裝好，且都以一整張圖的色素下去做處理，無法單以某個色素區塊去作處理。

**解決方法：**由公式去推倒，並撰寫新的演算法，即可分為八等分去作處理。

* **問題三：無法明確的抓取座標**

**造成原因：**在Picture Box中以Zoon顯示的話，所抓取到的座標跟實際上圖象的座標是錯誤的，會造成ROI所選取出來的圖像錯誤。

**解決方法：**在讀圖時先將圖檔縮小，並Picture Box以Auto Size的方式呈現，使得座標不會跑掉。

1. 結論

從Visual C#、AForge到Emgu的摸索花費了很長的時間，我們運用從簡單的一些程式語法到學習圖象的處理進行更深一步的了解，在撰寫程式當中可以不斷的刺激自己，該如何把程式編輯的更為簡單，將圖像處理的時間縮短，從原本熟悉的C、Java中跳脫到新的程式語法，這些都是以前我們所不知道的，也藉著實驗獲得了很好的經驗。

　　本專題雖然已完成基本架構，包常見的影像處理、ROI的撰寫，但仍然有不足之處，像ROI的固定單一圖形，僅能以圖像的方式作處理，較細密的影像處理，以上這些是我們正在努力去突破的困境。

　　我們也可以從競賽來了解此專題哪邊不足之處，與其他參賽選手交流從中獲得寶貴經驗，來不斷改進加強缺點，讓它能更趨於完美，總結醫療圖像處理專題讓我們收穫甚多，不論是增強自己撰寫程式的能力或者將此專題用於醫療上，這些都是以前沒有的，所以我們很高興能加入影像處理這團隊，希望未來影像處理能更加茁壯。

1. 參考資料

[1] 維基百科。醫學影像。網址：http://zh.wikipedia.org/wiki/%E9%86%AB%E5%AD%B8%E5%BD%B1%E5%83%8F。

[2] 石明于、陳耀添、陳怡萍。醫學影像處理與分割。網址：http://ip.csie.ncu.edu.tw/research/research-c05.htm。

[3]鄭俊彥。醫學影像處理。http://140.115.156.251/vclab/html/course/DIP%20Proposal/2009FallDIP\_983211009.pdf。

[4] 江青芬。2014。從2D到3D的醫學影像。科學發展503期。http://ejournal.stpi.narl.org.tw/NSC\_INDEX/Journal/EJ0001/10311/10311-03.pdf。

[6] Microsoft。https://msdn.microsoft.com/zh-tw/library/z1zx9t92.aspx。

[7] AForge。https://code.google.com/p/aforge/。

[8]Emgu 。http://www.emgu.com/wiki/index.php/Main\_Page。

1. 參加競賽成果

****

1. 參賽證明
2. 專案管理

**工作分配表：**

|  |  |  |  |
| --- | --- | --- | --- |
| 隊員姓名 | 初步工作分配(日期) | 工作調整(日期) | 實際工作情況 |
| 陳佳萱 | 程式編撰(3月20日) | 程式編撰(7月20日) | 文獻收集、程式編撰、書面整理。 |

**經費分配管理表：**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 經費編列項目 | | 規格(用途) | 單價 | 數量 | 總價(元) |
| 材料費 |  |  |  |  | 0 |
| 委外加工費 |  |  |  |  | 0 |
| 印刷費 | 文書資料 |  | 150 | 1 | 150 |
| 比賽海報 |  | 700 | 1 | 700 |
|  |  |  |  |  |
| 總金額(元) | 材料費 | 委外加工費 | 印刷費 | | 合計金額 |
| 0 | 0 | 850 | | 850 |

**工作時程規劃表**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 日期  工作  項目 | 103年01月 | 103年02月 | 103年03月 | 103年04月 | 103年05月 | 103年06月 | 103年07月 | 103年08月 |
| 資料收集 |  |  |  |  |  |  |  |  |
| 程式設計 |  |  |  |  |  |  |  |  |
| 實際測試 |  |  |  |  |  |  |  |  |
| 書面撰寫 |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 日期  工作  項目 | 103年09月 | 103年10月 | 103年11月 | 103年12月 | 103年01月 | 103年02月 | 103年03月 | 103年04月 |
| 資料收集 |  |  |  |  |  |  |  |  |
| 電路設計 |  |  |  |  |  |  |  |  |
| 實際測試 |  |  |  |  |  |  |  |  |
| 書面撰寫 |  |  |  |  |  |  |  |  |

**工作日誌**

|  |  |  |
| --- | --- | --- |
| **日期** | **工作概述** | **參與組員** |
| 103年01月25日 | 討論專題的主題。 | 陳佳萱 |
| 103年02月02日 | 文獻的收集。 | 陳佳萱 |
| 103年02月16日 | 討論要用什麼樣的程式去編撰。 | 陳佳萱 |
| 103年03月20日 | 學習Visual C#程式語法。 | 陳佳萱 |
| 103年04月02日 | 討論額外所需要擴充的軟件。 | 陳佳萱 |
| 103年04月08日 | 訂定了擴充的軟件AForge。 | 陳佳萱 |
| 103年04月24日 | Visual C#程式的學習以及其他文獻的參考。 | 陳佳萱 |
| 103年05月28日 | AForge的語法學習。 | 陳佳萱 |
| 103年05月30日 | 基本的使用介面設計出來。 | 陳佳萱 |
| 103年06月10日 | 將AForge與C#作程式的結合及撰寫。 | 陳佳萱 |
| 103年06月28日 | 發現所需要的功能不夠。 | 陳佳萱 |
| 103年07月08日 | 討論還要參考什麼樣的軟件，最後決定使用Emgu。 | 陳佳萱 |
| 103年07月12日 | Emgu的語法學習。 | 陳佳萱 |
| 103年08月16日 | 將基本功能以Visual C#及AForge完成。 | 陳佳萱 |
| 103年08月29日 | Emgu加入參考後，發現AForge與Emgu的變數格式不一樣。 | 陳佳萱 |
| 103年09月16日 | 修改程式語法。 | 陳佳萱 |
| 103年09月30日 | 完成後發現執行上會有些許Bug的存在，再次進行程式的修改。 | 陳佳萱 |
| 103年10月14日 | 收集許多的圖片來測試。 | 陳佳萱 |
| 103年10月28日 | 成品的完成。 | 陳佳萱 |
| 103年11月5日 | 參加學校創業比賽。 | 陳佳萱 |
| 103年12月6日 | 參加校慶專題參覽。 | 陳佳萱 |
| 103年12月7日 |
| 103年12月15日 | 開始編輯畢業專題書面資料。 | 陳佳萱 |
| 104年12月27日 | 智慧電子的報告書寫。 | 陳佳萱 |
| 104年01月28日 | 智慧電子的相關資料處理。 | 陳佳萱 |
| 104年04月10日 | 報告完成。 | 陳佳萱 |

1. 電子系學生實務專題(二)評量尺規(Rubrics)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **核心能力** | **權重%** | **典範(≥ 90)** | **優良(89~75)** | **尚可(74~60)** | **需再輔導(< 60)** | **評分** |
| 1.具備運用電子專業知識解決工程問題之能力。 | 20 | 能清楚且完整說明出專題的內容與理論基礎、研究動機、實驗結果與討論，並做出具體的結論。 | 能清楚且有條理地說明專題的內容、研究動機、實驗結果、成果討論與結論。 | 能說明專題的內容、研究動機、實驗結果、成果討論與結論。 | 無法清楚地說明專題的內容、研究動機、實驗結果、成果討論與結論。 |  |
| 2.具備操作電子工程相關系統及數據分析與詮釋之能力。 | 20 | 能使用儀器量測專題的所有參數值，將實驗數據圖表化，並作深入的分析及詮釋。 | 使用儀器量測專題的大部分參數值，將實驗數據圖表化，並作表淺的分析及詮釋。 | 使用儀器量測專題的少部分參數值，直接使用原始的實驗數據做表淺的說明。 | 未量測專題的參數值或缺乏實驗數據分析。 |  |
| 3.執行工程實務所需之電腦與資訊能力。 | 10 | 能繪製專業的電路圖及機構圖，且能撰寫高性能的專題驅動及控制程式，並能撰寫排版美觀的成果報告書、簡報及海報等。 | 能繪製專業的電路圖，且能撰寫正確的專題驅動及控制程式，並能撰寫排版美觀的成果報告書、簡報及海報等。 | 能撰寫大致正確的專題驅動及控制程式，及能用電腦軟體撰寫成果報告書、簡報及海報等。 | 未能撰寫大致正確的專題驅動及控制程式，或未能善用電腦軟體撰寫成果報告書、簡報及海報等。 |  |
| **4.**具備選擇及整合元件，改善或創新電子系統之能力，並對智慧財產權有所認知。 | 20 | 能於專題中的正確的選擇元件使作品性能最佳化，並能於報告書中說明理由及適當地引用參考文獻。 | 能於專題中適當的選擇元件改進作品性能，並能於報告書中說明理由及適當地引用參考文獻。 | 依循前人作品選擇元件，亦未於報告書中說明選擇元件的理由，但尚能適當地引用參考文獻。 | 元件選擇不恰當，或報告書涉嫌抄襲。 |  |
| 6.具備主動服務之熱誠、面對挫折抗壓性高之能力。 | 10 | 參與專題製作能完成團隊交付工作，並能積極主動協助他人。.參加競賽表現穩健，不患得患失。 | 參與專題製作能完成團隊交付工作，參加競賽表現穩健。 | 參與專題製作能完成團隊交付工作，參加競賽表現略微緊張。 | 參與專題製作未能完成團隊交付工作，或參加競賽表現失常。 |  |
| 7.發掘、分析、應用研究成果及因應複雜且整合性工程問題的能力。 | 10 | 專題作品涵蓋4個以上的工程領域，且專題作品參賽獲獎。 | 專題涵蓋3個以上的工程領域，且專題優於學長或同儕的作品。 | 專題涵蓋2個以上的工程領域，且專題能正常動作。 | 專題未能涵蓋2個以上的工程領域，或專題未能正常動作。 |  |
| 8.專案管理、有效溝通、領域整合與團隊合作的能力。 | 10 | 團隊能做好專案規劃，且能同心協力、相互支援，依規劃內容完成專題作品。 | 團隊能依專案規劃，各自分工，完成專題作品。 | 實際工作與專案規劃內容不盡相符，但仍能完成專題作品。 | 團隊缺乏向心力，專案規劃內容不切實際。 |  |
| 總分 |  | | | | | |

指導老師：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_